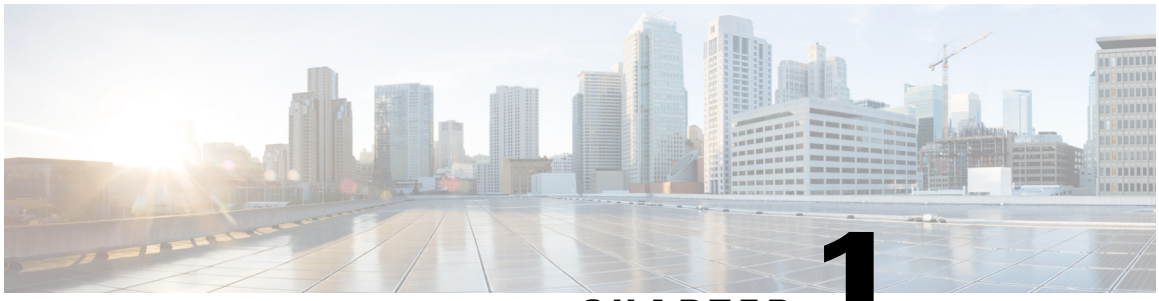
Cisco Media Services Proxy Configuration Guide, Cisco IOS Release 15M&T

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Media Services Proxy

Media Services Proxy (MSP) is one of the features of the Medianet Media Awareness capability. MSP makes the network intelligent by automatically identifying various media endpoints and rendering media services such as admission control, flow metadata, and auto smart ports accordingly. It acts as a layer that automatically connects devices with their respective network services.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Media Services Proxy

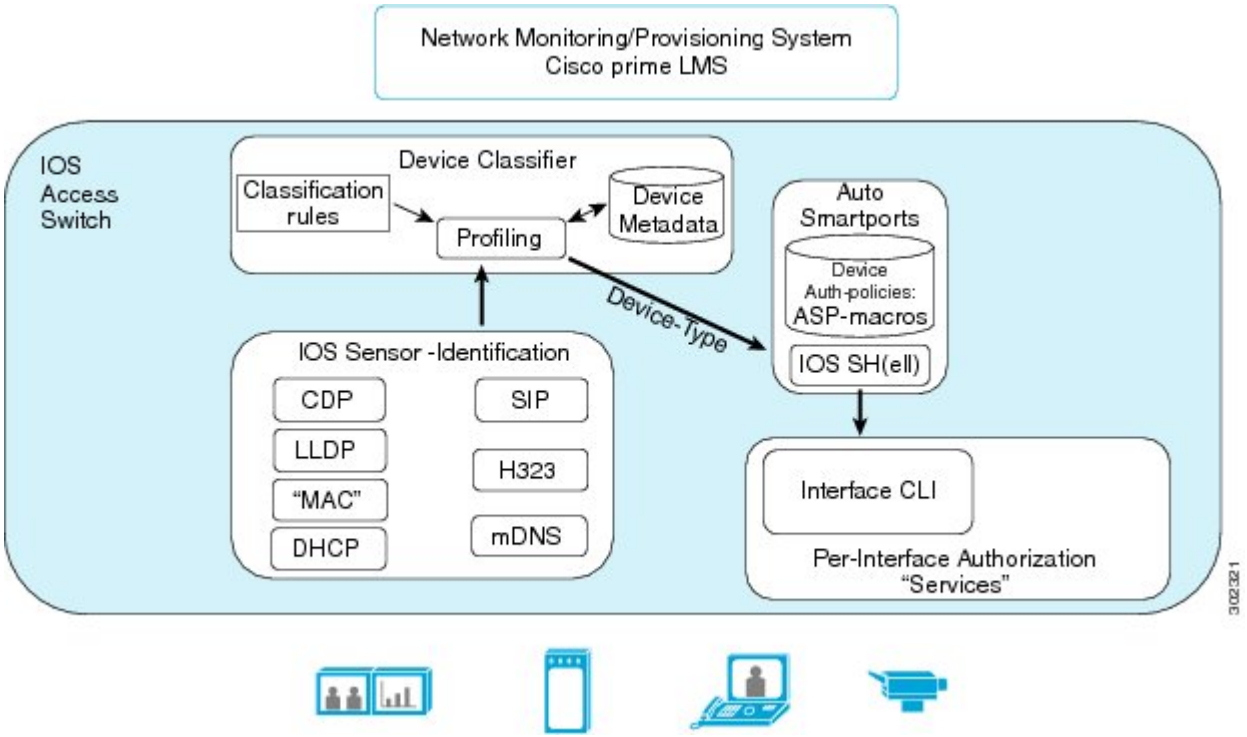
- Device and flow identification are not IPv6 compatible.
- Media monitoring as a service is not available in Catalyst 4500 series switches.

Information About Media Services Proxy

With a growing number of media endpoints, the network must understand and provide appropriate media services to the endpoints. Following are some of the basic services that a typical endpoint requires:

- Device Identification and characterizations
- Flow metadata signaling for network services such as quality of services (QoS) and call admission control (CAC)

MSP follows a network-centric model, where access switches and routers learn information about devices and flow automatically. The figure below shows a high level view of device and flow identification mechanism used by MSP. The figure below illustrates the interaction of Cisco IOS device sensor framework with the device classifier to identify the device type. The Cisco IOS device sensor feature gleans endpoint device information from protocols such as Cisco Discovery Protocol, Link Layer Discovery Protocol (LLDP), and Dynamic Host Control Protocol (DHCP). MSP leverages the Cisco IOS device sensor framework to glean information from additional protocols such as Multicast Domain Name System (mDNS), H323, and Session Initiation Protocol (SIP).



Based on the type of device identified, the physical interface to which the device is connected can be configured using auto smart ports with minimal configuration by network administrator. Based on the types of media flows identified, MSP provides services such as CAC and QoS to the network devices. The Device Identification and Flow Identification Mechanisms sections provide more information about the mechanisms used to identify devices and media flows.

Benefits of MSP

Following are the benefits of MSP:

- Automatic identification of devices and media flow in the network.
- Automatic application of appropriate services to the endpoints.
- Configuration control for the administrator, thereby reducing the manual configuration and management of services.

Device Identification

MSP leverages the Cisco IOS device sensor infrastructure to facilitate device identification and classification. Device sensor provides device identification for media endpoints through Cisco Discovery Protocol, DHCP, and LLDP. MSP aids in device identification through additional protocols such as mDNS, H.323, and SIP. Video conference systems use H.323 and SIP control packets for voice or video call setup. IP cameras use mDNS control packets to register or exchange initial control information with the surveillance manager.

You can use the **profile flow** command to enable MSP, which automatically enables the device identification and classification on the access switch or router. Use the **show profile device** command to view the devices that are automatically identified.

How Does Device Identification and Classification Work

Device identification occurs by extracting the raw endpoint data from the network devices. The endpoint information that is gathered aids in completing the profiling capability of devices. Profiling is the determination of the endpoint type based on information gleaned from various protocol packets from an endpoint during its connection to a network. The profiling capability consists of two parts:

- Collector—Gathers endpoint data from the endpoint network devices through protocols such as Cisco Discovery Protocol, LLDP, and DHCP subject to statically configured filters, and makes this information available to its registered clients.
- Analyzer—Processes the data and determines the type, model, and class of the device. The analyzer is either embedded within IOS or by using an external device called Positron.

The endpoint device has its own lifecycle from the time it comes up till the time it goes down, which is managed using a session manager. One session is created per endpoint device attached to the network element. The session manager interfaces with the device classifier to analyze the information collected. The device classifier is a collection of rules that are applied to the device metadata attributes. Device metadata attributes are evaluated against a set of profiles available to the device classifier to determine the best match. Based on the best-matched profile, the device type is determined, thus creating device visibility. Device visibility helps in understanding the ongoings of the network, without actually impacting the network unless the network administrator prefers.

In the MSP feature, media endpoints are identified by parsing initial control packet exchange between the endpoints or the media server. These control packets are copied by MSP and original packets are forwarded to the destination. MSP parses the protocol packets and derives type, length, values (TLV) tables. These TLV tables are used to identify the media endpoints.

For more information on device identification media endpoints through device sensor, refer to the Device Sensor Configuration Guide.

Device Services

Based on the type of device identified, you can choose to configure auto smart ports.

Auto Smartports macros dynamically configure ports based on the device type detected on the port. When the access switch or router detects a new device on a port, it applies the appropriate macro on the port. When there is a link-down event on the port, the macro is removed. For example, when you connect a Cisco IP phone to a port, Auto Smartports automatically applies the IP phone macro. The IP phone macro enables quality of service (QoS), security features, and a dedicated voice VLAN to ensure proper treatment of delay-sensitive voice traffic. Auto Smartports uses event triggers to map devices to port macros.

You can also manually configure and apply global macros. The macros embedded in the Cisco device software are groups of command-line interface (CLI) commands.

You can also create user-defined macros by using the Cisco IOS Shell scripting capability, which is a BASH-like language syntax for command automation and variable replacement.

For more information on configuring auto smart ports, see Auto Smart Port Configuration Guide.

Flow Identification

MSP facilitates automatic identification of media flows by using protocols such as SIP, H.323, and RTSP. MSP maintains a database of 5-tuple media flow and associated flow metadata attributes (such as application type, vendor, version, and audio or video media type) after flow identification. These metadata attributes are used to classify the types of media flows and render media services such as CAC and QoS.

You can use the **profile flow** command to enable MSP, which automatically enables flow identification on the access router or switch. The **show profile flow** command displays all the media flows that have been automatically identified.

Flow Services

After the media flows have been identified, the 5-tuple flow identifier and the associated flow metadata attributes that have been extracted out of protocol exchange are stored in the metadata database. These attributes can be used to provide network services such as RSVP CAC and QoS.

MSP derives the desired media bandwidth from the initial protocol exchange between the endpoints. You can also manually configure RSVP bandwidth, which overrides the bandwidth that is automatically identified. When RSVP signaling is configured as part of MSP, access routers or switches generate RSVP packets for bandwidth reservation and forward them to the downstream routers. Actual bandwidth reservation or CAC is carried over at the downstream routers that are connected to the access routers or switches. Catalyst 4500 series do not support RSVP CAC.

Quality of services such as controlling, policing, classification, and marking can be provided to the automatically identified media flows by using metadata attributes extracted from the media flows.

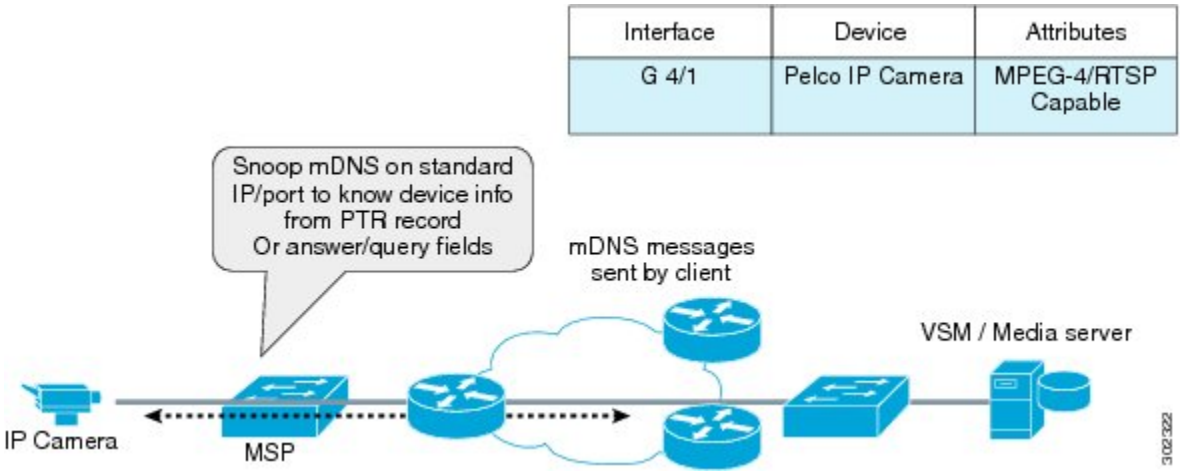
You can use EEM scripts to contain the required flow services to be applied on specific media flows. MSP flow services are applied to the network devices that are directly connected to the Layer 2 (L2) physical interfaces of the media endpoints. You can also create MSP profiles containing the required services to be applied to the flow globally or on a per-interface basis.

For instance, you can create an MSP profile where a SIP flow matching payload type 96, bandwidth of 64 kb/s, and an audio codec of G.711 can initiate RSVP bandwidth reservations.

Device Identification Mechanisms

mDNS Based Device Discovery

The following figure shows the mDNS device discovery mechanism.



The figure shows an IP camera connected to the networking device (on which MSP is enabled). The IP camera sends mDNS messages to the multicast IP address 224.0.0.251 on standard mDNS port 5353. The networking device listens to these messages on the standard mDNS port and derives the device type and class. Based on these attribues, the device classifier looks up the best match and completes the profiling. Following is sample packet capture, which highlights the device name and class.

```
Frame 48: 561 bytes on wire (4488 bits), 561 bytes captured (4488 bits)
Ethernet II, Src: AxisComm_ad:c9:93 (00:40:8c:ad:c9:93), Dst: IPv4mcast_00:00:fb
(01:00:5e:00:00:fb)
Internet Protocol Version 4, Src: 10.254.148.190 (10.254.148.190), Dst: 224.0.0.251
(224.0.0.251)
User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
Domain Name System (response)
    [Request In: 45]
    [Time: 1.290247000 seconds]
    Transaction ID: 0x0000
    Flags: 0x8400 (Standard query response, No error)
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .1.. .... .... = Authoritative: Server is an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...0 .... .... = Recursion desired: Don't do query recursively
        .... .... 0... .... = Recursion available: Server can't do recursive queries
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..0. .... = Answer authenticated: Answer/authority portion was not
authenticated by the server
        .... .... ...0 .... = Non-authenticated data: Unacceptable
        .... .... .... 0000 = Reply code: No error (0)
    Questions: 0
    Answer RRs: 16
    Authority RRs: 0
    Additional RRs: 0
    Answers
        axis-00408cadc993.local: type A, class IN, cache flush, addr 10.254.148.190
       10.148.254.169.in-addr.arpa: type PTR, class IN, cache flush, axis-00408cadc993.local
```

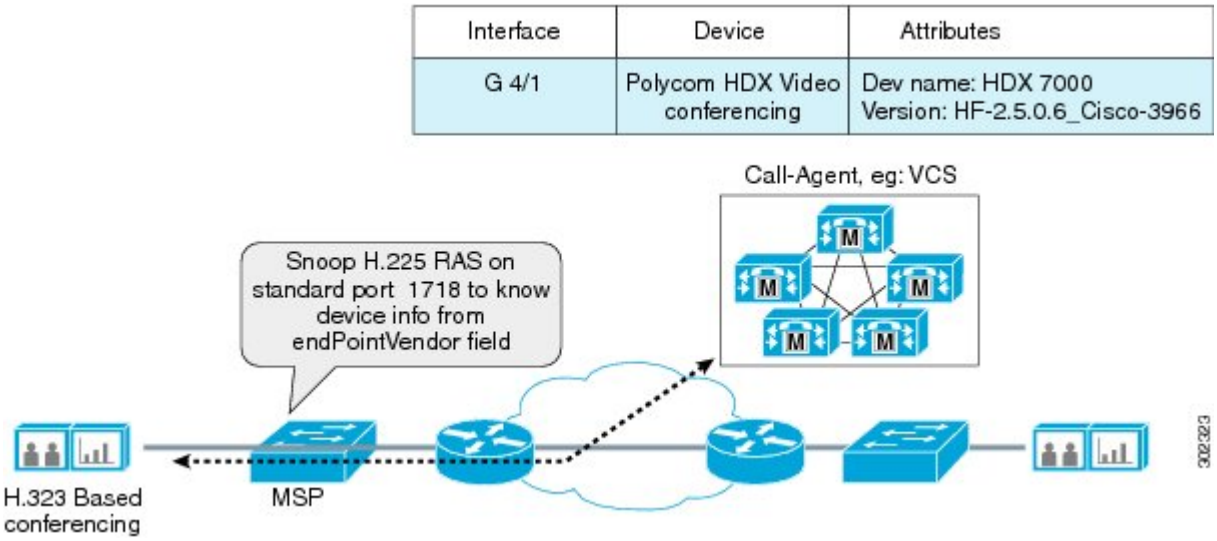
```
        axis-00408cadc993.local: type A, class IN, cache flush, addr 192.168.0.90
       90.0.168.192.in-addr.arpa: type PTR, class IN, cache flush, axis-00408cadc993.local

      AXIS M1114 - 00408CADC993._http._tcp.local: type SRV, class IN, cache flush, priority
 0, weight 0, port 80, target axis-00408cadc993.local

No.     Time        Source                Destination           Protocol Length Info
     49 31.063350   0.0.0.0               255.255.255.255       DHCP     590    DHCP Discover
 - Transaction ID 0x2a5dad4a
```

H.323-Based Device Discovery

The following figure shows the H.323 device based discovery.



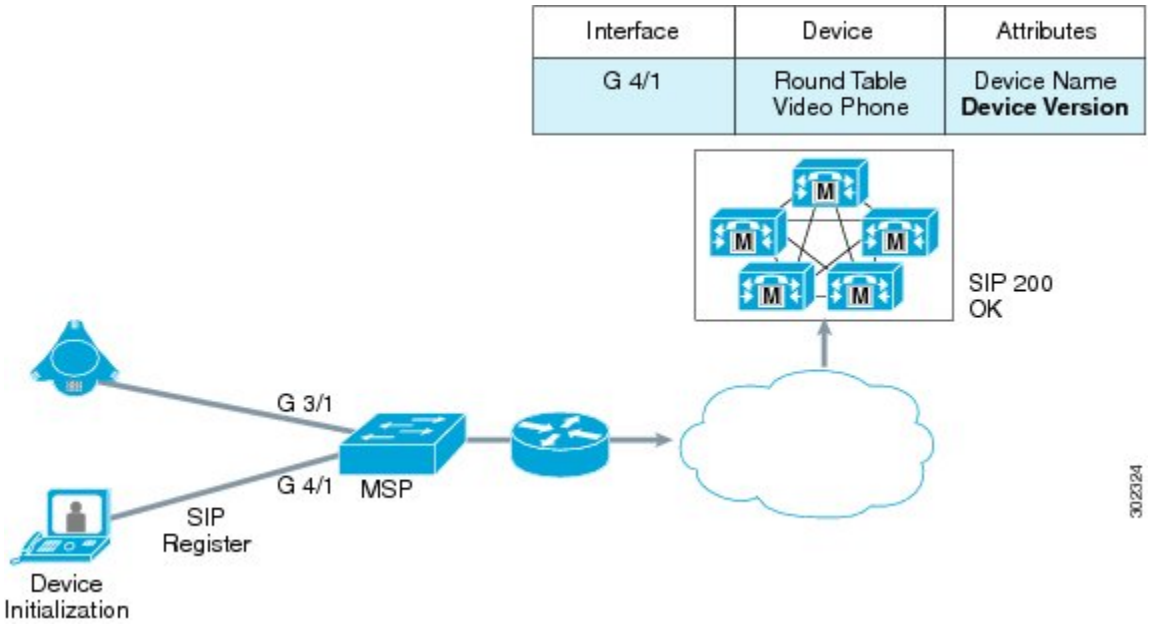
The H.323 client, which is a video conferencing system sends H.225 RAS client registration message to the call agent. The networking device (on which MSP is enabled) snoops H.225 messages on the standard port 1718 to interpret the device information. Following sample packet capture highlights the Vendor field in H.225 messages, which identifies the device class, vendor, and version details. The device classifier uses the device class, vendor, and version details to profile the device accordingly.

```
Frame 53: 266 bytes on wire (2128 bits), 266 bytes captured (2128 bits)
Ethernet II, Src: Viavideo_0c:99:c7 (00:e0:db:0c:99:c7), Dst: Cisco_44:b4:bf
(d0:d0:fd:44:b4:bf)
Internet Protocol Version 4, Src: 10.0.0.100 (10.0.0.100), Dst: 10.0.0.101 (10.0.0.101)
Transmission Control Protocol, Src Port: 49152 (49152), Dst Port: h323hostcall (1720), Seq:
 1, Ack: 1, Len: 200
TPKT, Version: 3, Length: 200
Q.931
H.225.0 CS
    H323-UserInformation
        h323-uu-pdu
            h323-message-body: setup (0)
                setup
                    protocolIdentifier: 0.0.8.2250.0.4 (Version 4)
                    sourceAddress: 2 items
                        Item 0
                            AliasAddress: h323-ID (1)
                                h323-ID: Polycom2
                        Item 1
                            AliasAddress: h323-ID (1)
                                h323-ID: Polycom2
                    sourceInfo
                        vendor
                            vendor
```

```
                                t35CountryCode: United States (181)
                                t35Extension: 0
                                manufacturerCode: 9009
                            H.221 Manufacturer: ViaVideo (0xb5002331)
                            productId: HDX 7000
                            versionId: HF.2.5.0.6_Cisco-3966
                        terminal
                        ..0. .... mc: False
                        ...0 .... undefinedNode: False
                    destCallSignalAddress: ipAddress (0)
                    0... .... activeMC: False
                    conferenceID: 02344ebe-3c00-1000-1ed2-c9ceeffc85db
                    conferenceGoal: create (0)
                    callType: pointToPoint (0)
                    sourceCallSignalAddress: ipAddress (0)
                    callIdentifier
                    0... .... mediaWaitForConnect: False
                    0... .... canOverlapSend: False
                    0... .... multipleCalls: False
                    0... .... maintainConnection: False
                    presentationIndicator: presentationAllowed (0)
                        presentationAllowed: NULL
                    screeningIndicator: userProvidedVerifiedAndFailed (2)
            0... .... h245Tunnelling: False
    user-data
```

SIP-Based Device Discovery

The following figure shows the SIP-based device discovery.



The SIP client, which is a round table video phone sends out SIP Register messages to the call manager. The call manager is responsible for routing the call across the enterprise network. Following sample packet capture highlights the UserAgent field in the SIP Register message, which identifies the device name and the device version.

The device classifier uses the device name and version details to profile the device accordingly.

```
Frame 24: 602 bytes on wire (4816 bits), 602 bytes captured (4816 bits)
Ethernet II, Src: Viavideo_0c:96:de (00:e0:db:0c:96:de), Dst: Cisco_f7:12:00
(d0:d0:fd:f7:12:00)
Internet Protocol Version 4, Src: 10.0.0.95 (10.0.0.95), Dst: 10.1.1.4 (10.1.1.4)
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
```

```
Session Initiation Protocol
    Request-Line: REGISTER sip:10.1.1.4 SIP/2.0
        Method: REGISTER
        Request-URI: sip:10.1.1.4
            Request-URI Host Part: 10.1.1.4
        [Resent Packet: False]
    Message Header
        Via: SIP/2.0/UDP 10.0.0.95:5060;branch=z9hG4bK10048000-287329697
            Transport: UDP
            Sent-by Address: 10.0.0.95
            Sent-by port: 5060
            Branch: z9hG4bK10048000-287329697
        Max-Forwards: 70
        Allow:
INVITE,BYE,CANCEL,ACK,INFO,PRACK,COMET,OPTIONS,SUBSCRIBE,NOTIFY,REFER,REGISTER,UPDATE
        Supported: ms-forking,replaces
        From: 1020 <sip:1020@10.1.1.4> ;epid=8210200C96DECN;tag=plcm_10050000-287329698
            SIP Display info: 1020
            SIP from address: sip:1020@10.1.1.4
                SIP from address User Part: 1020
                SIP from address Host Part: 10.1.1.4
            SIP tag: plcm_10050000-287329698
        To:  <sip:1020@10.1.1.4>
            SIP to address: sip:1020@10.1.1.4
                SIP to address User Part: 1020
                SIP to address Host Part: 10.1.1.4
        Call-ID: 10047000-287329696
        CSeq: 1 REGISTER
            Sequence Number: 1
            Method: REGISTER
        Expires: 300
        Contact: 1020 <sip:1020@10.0.0.95:5060;transport=udp> ;proxy=replace
            SIP Display info: 1020
            Contact-URI: sip:1020@10.0.0.95:5060;transport=udp
                Contactt-URI User Part: 1020
                Contact-URI Host Part: 10.0.0.95
                Contact-URI Host Port: 5060
            Contact parameter: transport=udp>
            Contact parameter: proxy=replace
        User-Agent: Polycom HDX 7000 (HF - 2.5.0.6_00_Cisco-3966)
        Content-Length: 0

No.     Time        Source                Destination           Protocol Length Info
     25 29.812516   10.0.0.95             10.1.1.4              SIP      602    Request:
REGISTER sip:10.1.1.4
```

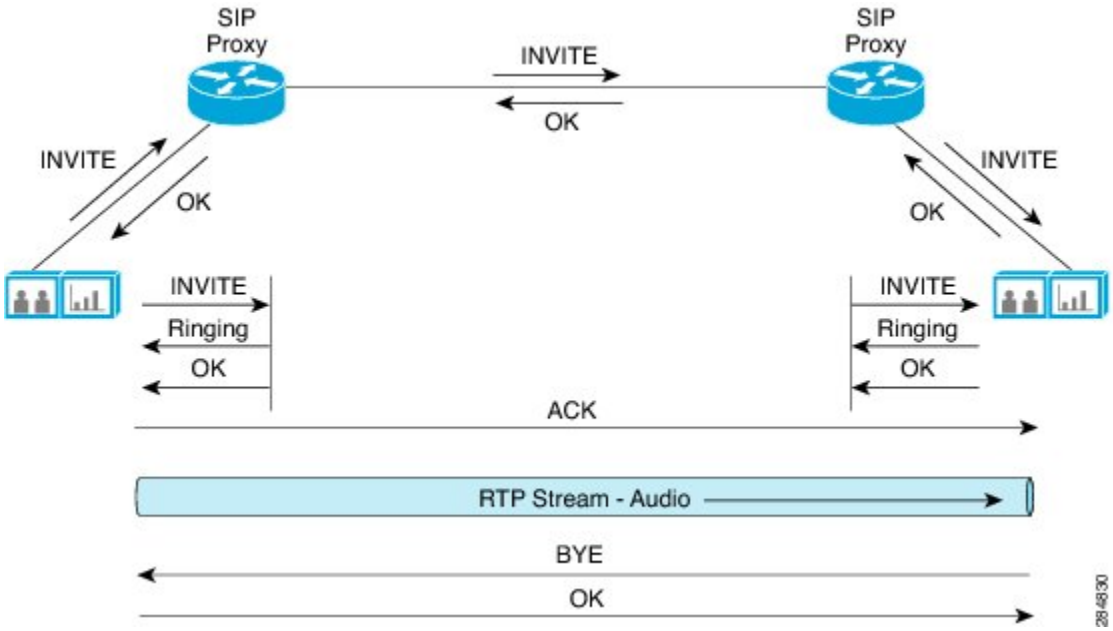
Flow Identification Mechanisms

SIP-Based Flow Identification

The SIP is an application-level signaling protocol used for controlling multimedia communication sessions such as voice and video calls. SIP enables one party to place a call to another party and negotiates the parameters of a multimedia session. The actual audio, video, or other multimedia content is exchanged between session participants using the Real-Time Transport Protocol (RTP).

SIP incorporates the use of a Session Description Protocol (SDP), which defines the session content. SIP is used to invite one or more participants to a session, and the SDP-encoded body of the SIP message contains information about what media encoding (for example, voice or video) the parties use.

The figure below displays the SIP message exchange process.



In the message exchange process for SIP, the INVITE message, which is used to establish a media session between user agents, carries SDP from the sender to the receiver. The associated SDP provides information about the bandwidth, the application name, and the sender port number. It also signals RTP, which is used as the protocol for communications.

The receiver sends the OK response along with the SDP, which includes the audio port of the destination. The complete 5-tuple information is derived from the INVITE and the OK message exchanges. This information is then used by flow metadata or RSVP proxy to provide the necessary services in the forward direction of the RTP flow.

The following sample packet captures display the SIP message exchange process. The text highlighted in bold indicates the 5-tuple information that are extracted by flow metadata.

```
Frame 1216: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits)
Ethernet II, Src: Viavideo_0c:96:de (00:e0:db:0c:96:de), Dst: Cisco_f7:12:00
(d0:d0:fd:f7:12:00)
Internet Protocol Version 4, Src: 10.0.0.95 (10.0.0.95), Dst: 10.1.1.4 (10.1.1.4)
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
Session Initiation Protocol
    Request-Line: INVITE sip:1009@10.1.1.4 SIP/2.0
        Method: INVITE
        Request-URI: sip:1009@10.1.1.4
        [Resent Packet: False]
    Message Header
        Via: SIP/2.0/UDP 10.0.0.95:5060;branch=z9hG4bK51903000-287329707
        Max-Forwards: 70
        From: 1020 <sip:1020@10.1.1.4> ;epid=8210200C96DECN;tag=plcm_51345000-287329705
        To:  <sip:1009@10.1.1.4>
        Call-ID: 51344000-287329703
        CSeq: 3 INVITE
        Min-SE: 1800
        Session-Expires: 1800
        Supported: ms-forking,timer
        Contact: 1020 <sip:1020@10.0.0.95:5060;transport=udp> ;proxy=replace
        Content-Type: application/sdp
        Authorization: Digest
username="1020@10.1.1.4",realm="cmsipline",nonce="bC2sMqqTXlsQlSZVPiIkUhkEfUyF",uri="sip:1009@10.1.1.4",response="a7d7e99b96e6d14e04ae2a0c092b3",algorithm=MD5

        User-Agent: Polycom HDX 7000 (HF - 2.5.0.6_00_Cisco-3966)
        Content-Length: 826
    Message Body
        Session Description Protocol
```

```
            Session Description Protocol Version (v): 0
            Owner/Creator, Session Id (o): bangalore 1804537739 0 IN IP4 10.0.0.95
            Session Name (s): -
            Connection Information (c): IN IP4 10.0.0.95
            Bandwidth Information (b): CT:1920
            Time Description, active time (t): 0 0
            Media Description, name and address (m): audio 49154 RTP/AVP 115 102 9 15 0 8
18 119
            Media Attribute (a): rtpmap:115 G7221/32000
            Media Attribute (a): fmtp:115 bitrate=48000
            Media Attribute (a): rtpmap:102 G7221/16000
            Media Attribute (a): fmtp:102 bitrate=32000
            Media Attribute (a): rtpmap:9 G722/8000
            Media Attribute (a): rtpmap:15 G728/8000
            Media Attribute (a): rtpmap:0 PCMU/8000
            Media Attribute (a): rtpmap:8 PCMA/8000
            Media Attribute (a): rtpmap:18 G729/8000
            Media Attribute (a): fmtp:18 annexb=no
            Media Attribute (a): rtpmap:119 telephone-event/8000
            Media Attribute (a): fmtp:119 0-15
            Media Attribute (a): sendrecv
            Media Description, name and address (m): video 49156 RTP/AVP 109 96 34 31
            Bandwidth Information (b): TIAS:384000
            Media Attribute (a): rtpmap:109 H264/90000
            Media Attribute (a): fmtp:109 profile-level-id=42800d; max-mbps=47520;
max-fs=1584; max-br=1600; sar=13
            Media Attribute (a): rtpmap:96 H263-1998/90000
            Media Attribute (a): fmtp:96 CIF4=2;CIF=1;QCIF=1;SQCIF=1;F;J;T
            Media Attribute (a): rtpmap:34 H263/90000
            Media Attribute (a): fmtp:34 CIF4=2;CIF=1;QCIF=1;SQCIF=1;F
            Media Attribute (a): rtpmap:31 H261/90000
            Media Attribute (a): fmtp:31 CIF=1;QCIF=1
            Media Attribute (a): sendrecv
            Media Attribute (a): rtcp-fb:* ccm fir tmmbr
```

The first line of the message contains the method name (INVITE), the SIP Universal Resource Indicator (URI), and the version number.

The message header lists various details of the message including the content type that indicates the type of the message body.

The message body for this particular SIP message lists the contents of SDP such as bandwidth information, application name, and the clock frequency.

A sample OK message is as follows:

```
Ethernet II, Src: Cisco_f7:12:00 (d0:d0:fd:f7:12:00), Dst: Viavideo_0c:96:de
(00:e0:db:0c:96:de)
Internet Protocol Version 4, Src: 10.1.1.4 (10.1.1.4), Dst: 10.0.0.95 (10.0.0.95)
User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)
    Session Initiation Protocol
      Status-Line: SIP/2.0 200 OK
        Status-Code: 200
        [Resent Packet: False]
        [Request Frame: 53]
        [Response Time (ms): 16079]
    Message Header
        Via: SIP/2.0/UDP 10.0.0.95:5060;branch=z9hG4bK51903000-287329707
        From: 1020 <sip:1020@10.1.1.4> ;epid=8210200C96DECN;tag=plcm_51345000-287329705
        To: <sip:1009@10.1.1.4> ;tag=5bc4d0f5-acc3-43e2-a11d-3ea8aae3458a-25577575
        Date: Thu, 11 Nov 2010 15:12:08 GMT
        Call-ID: 51344000-287329703
        CSeq: 3 INVITE
        Allow: INVITE, OPTIONS, INFO, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE,
NOTIFY
        Allow-Events: presence
        Contact: <sip:1009@10.1.1.4:5060>
        Supported: replaces
        Send-Info: conference
        Session-Expires:  1800;refresher=uas
        Require:  timer
        Remote-Party-ID: <sip:1009@10.1.1.4>;party=called;screen=yes;privacy=off
```

```
    Content-Type: application/sdp
    Content-Length: 514
Message Body
    Session Description Protocol
        Session Description Protocol Version (v): 0
        Owner/Creator, Session Id (o): CiscoSystemsCCM-SIP 2000 1 IN IP4 10.1.1.4
        Session Name (s): SIP Call
        Time Description, active time (t): 0 0
        Media Description, name and address (m): audio 21426 RTP/AVP 9 101
        Connection Information (c): IN IP4 20.0.0.68
        Media Attribute (a): rtpmap:9 G722/8000
        Media Attribute (a): ptime:20
        Media Attribute (a): rtpmap:101 telephone-event/8000
        Media Attribute (a): fmtp:101 0-15
        Media Description, name and address (m): video 0 RTP/AVP 31 34 96 97
        Connection Information (c): IN IP4 0.0.0.0
        Media Attribute (a): rtpmap:31 H261/90000
        Media Attribute (a): fmtp:31 MAXBR=128
        Media Attribute (a): rtpmap:34 H263-1998/90000
        Media Attribute (a): fmtp:34 SQCIF=1;QCIF=1;CIF=1;CIF4=2;F=1;J=1;T=1
        Media Attribute (a): rtpmap:96 H263-1998/90000
        Media Attribute (a): fmtp:96 SQCIF=1;QCIF=1;CIF=1;CIF4=2;F=1
        Media Attribute (a): rtpmap:97 H264/90000
        Media Attribute (a): fmtp:97 parameter-add=0
        Media Attribute (a): inactive
```

The first line of the OK message contains the version number of SIP used and the 200 OK response code and name. The message header lists various details of the message, including the content type, which indicates the type of the message body.

The message body lists the contents of the SDP, which contains the audio port of the destination.

The sample 5-tuple derived from the message headers for the RTP session are listed in the table below.

Table 1: Tuple Values Derived from the Headers for the Sample SIP Session

Tuple	Values
Source IP	10.1.1.4
Destination IP	10.0.0.95
Source Port	5060
Destination Port	5060
Protocol	RTP

H.323-Based Flow Identification

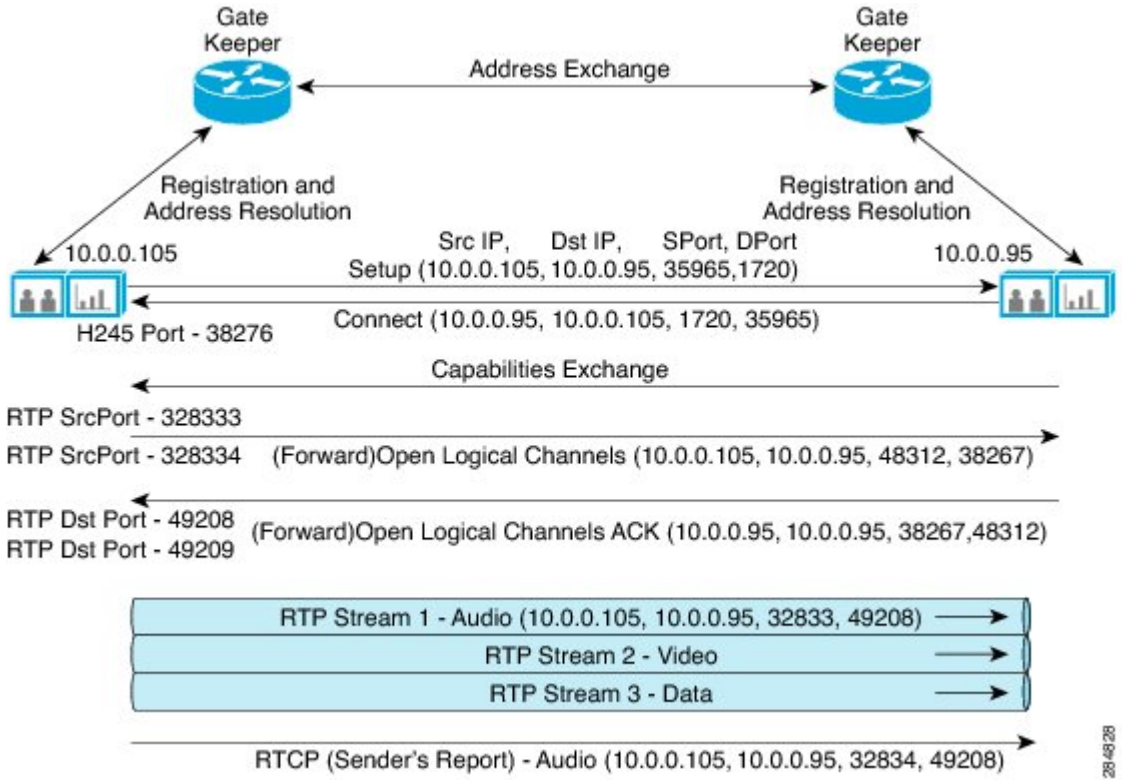
H.323 is a system specification that describes the use of several ITU-T and IETF protocols that provide audio, video, and data communications in any IP-based network.

The H.323 protocol suite is split into three main areas of control:

- Registration, Admission, and Status (RAS) (H.225) signaling—used between an H.323 endpoint and a gatekeeper to provide address resolution and admission control services.

- Call Control/Call Setup (H.225)—used between any two H.323 entities to establish communication. This happens over port 1720 and provides the necessary flow metadata required to establish CAC or a flow metadata session.
- H.245 Media Control and Transport Signaling—used for multimedia communication that describes the messages and procedures used for capability exchange, opening and closing logical channels for audio, video and data control, and indications. This happens in parallel to a separate TCP session but on a dynamic port.

The figure below shows a sample H.323 message exchange process. H.323 version 1 is used. The Catalyst 4500 series switches support only up to 13 simultaneous H.323 v1 calls.



The packet captures for the sample message exchange process and the corresponding flow metadata attributes extracted are described in the following section.

The process starts with the device discovery followed by registration. Gateways and terminals join a zone and inform their gatekeepers of their IP and alias addresses. This is followed by the H.225 call control signaling to set up connections between the two H.323 endpoints starting with a Setup message. A call control channel is created across an IP network on TCP port 1720. This port initiates the Q.931 call control messages for the purpose of the connection, maintenance, and disconnection of calls.

```
Ethernet II, Src: Viavideo_0c:96:de (00:e0:db:0c:96:de), Dst: Radvisio_01:14:93
(00:03:d6:01:14:93)
Internet Protocol Version 4, Src: 10.0.0.105 (10.0.0.105), Dst: 10.0.0.95 (10.0.0.95)
Transmission Control Protocol, Src Port: 35965 (35965), Dst Port: h323hostcall (1720), Seq:
 1, Ack: 1, Len: 242
TPKT, Version: 3, Length: 242
Q.931
H.225.0 CS
    H323-UserInformation
        h323-uu-pdu
            h323-message-body: setup (0)
                setup
```

```
                protocolIdentifier: 0.0.8.2250.0.4 (Version 4)
                sourceAddress: 2 items
                sourceInfo
                    vendor
                        vendor
                            t35CountryCode: United States (181)
                            t35Extension: 0
                            manufacturerCode: 9009
                        H.221 Manufacturer: ViaVideo (0xb5002331)
                        productId: HDX 7000
                        versionId: HF - 2.5.0.6_00_Cisco-3966
                    terminal
                    ..0. .... mc: False
                    ...0 .... undefinedNode: False
                destinationAddress: 1 item
                destCallSignalAddress: ipAddress (0)
                    ipAddress
                        ip: 10.0.0.95 (10.0.0.95)
                        port: 1720
                0... .... activeMC: False
                conferenceID: 02324671-8f87-1140-1312-7b98f1d65745
                conferenceGoal: create (0)
                callType: pointToPoint (0)
                sourceCallSignalAddress: ipAddress (0)
                    ipAddress
                        ip: 10.0.0.105 (10.0.0.105)
                        port: 35965
                callIdentifier
                0... .... mediaWaitForConnect: False
                0... .... canOverlapSend: False
                endpointIdentifier: 206D3CB800000002
                0... .... multipleCalls: False
                0... .... maintainConnection: False
                presentationIndicator: presentationAllowed (0)
                screeningIndicator: userProvidedVerifiedAndFailed (2)
            0... .... h245Tunnelling: False
        user-data
```

The following table lists the flow metadata attributes derived from the headers of the H.225 call control signaling for the sending device.

Table 2: Flow Metadata Attributes Derived from the Headers for the Sample H.323 Session (Source)

Flow Metadata Attributes	Values
Source Model	HDX 7000
Source Version	HF - 2.5.0.6_00_Cisco-3966
Source IP	10.0.0.105
Source Port	35965
H.245 Tunneling	FALSE
Destination IP	10.0.0.95
Destination Port	1720

The following sample packet capture displays flow metadata attributes for the receiving device:

```
H.225.0 CS
    H323-UserInformation
```

```
h323-uu-pdu
    h323-message-body: connect (2)
        connect
            protocolIdentifier: 0.0.8.2250.0.5 (Version 5)
            h245Address: ipAddress (0)
                ipAddress
                    ip: 10.0.0.105 (10.0.0.105)
                    port: 39161
            destinationInfo
                vendor
                    vendor
                        t35CountryCode: Italy (89)
                        t35Extension: 0
                        manufacturerCode: 44547
                    H.221 Manufacturer: viavideo (0xb5002331)
                    productId: RV XT1000
                    versionId: V1.0.19 Mon May 31 16:02:37 2010
                terminal
                ..0. .... mc: False
                ...0 .... undefinedNode: False
            conferenceID: 02324671-8f87-1140-1312-7b98f1d65745
            callIdentifier
                guid: 02324671-8f87-1140-1311-7b98f1d65745
            0... .... multipleCalls: False
            1... .... maintainConnection: True
            presentationIndicator: presentationAllowed (0)
                presentationAllowed: NULL
            screeningIndicator: userProvidedVerifiedAndFailed (2)
    0... .... h245Tunnelling: False
```

Table 3: Flow Metadata Attributes Derived from the Headers for the Sample H.323 Session (Receiver)

Flow Metadata Attributes	Values
Receiver Model	RV XT1000
Receiver Version	V1.0.19
H245 Tunneling	FALSE

The logical channels are established by the H.245 media control and transport signaling for transmitting audio, video, data, and control channel information. The channel usage and flow control capabilities are negotiated.

```
Frame 76: 111 bytes on wire (888 bits), 111 bytes captured (888 bits)
Ethernet II, Src: Viavideo_0c:96:de (00:e0:db:0c:96:de), Dst: Radvisio_01:14:93
(00:03:d6:01:14:93)
Internet Protocol Version 4, Src: 10.0.0.95 (10.0.0.95), Dst: 10.0.0.105 (10.0.0.105)
Transmission Control Protocol, Src Port: 35940 (35940), Dst Port: 39161 (39161), Seq: 619,
 TPKT, Version: 3, Length: 45
H.245
    PDU Type: request (0)
        request: openLogicalChannel (3)
            openLogicalChannel
                forwardLogicalChannelNumber: 2
                forwardLogicalChannelParameters
                    dataType: audioData (3)
                        audioData: genericAudioCapability (20)
                        genericAudioCapability
                        capabilityIdentifier: standard (0)
                        standard: 0.0.7.7221.1.1.0 (itu-t.0.7.7221.1.1.0)
                        maxBitRate: 480
                        collapsing: 2 items
                        Item 0
                        collapsing item
                        parameterIdentifier: standard (0)
                        standard: 1
                        parameterValue: unsignedMin (2)
```

```
                        unsignedMin: 1
                        Item 1
                        collapsing item
                        parameterIdentifier: standard (0)
                        standard: 2
                        parameterValue: booleanArray (1)
                        booleanArray: 16
                    multiplexParameters: h2250LogicalChannelParameters (3)
                        h2250LogicalChannelParameters
                            sessionID: 1
                            mediaControlChannel: unicastAddress (0)
                                unicastAddress: iPAddress (0)
                                    iPAddress
                                        network: 10.0.0.95 (10.0.0.95)
                                        tsapIdentifier: 49155
                            dynamicRTPPayloadType: 115

Frame 1045: 110 bytes on wire (880 bits), 110 bytes captured (880 bits)
Ethernet II, Src: Radvisio_01:14:93 (00:03:d6:01:14:93), Dst: Viavideo_0c:96:de
(00:e0:db:0c:96:de)
Internet Protocol Version 4, Src: 10.0.0.105 (10.0.0.105), Dst: 10.0.0.95 (10.0.0.95)
User Datagram Protocol, Src Port: filenet-rpc (32769), Dst Port: 49155 (49155)
Real-time Transport Control Protocol (Sender Report)
    [Stream setup by H245 (frame 83)]
        [Setup frame: 83]
        [Setup Method: H245]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 0001 = Reception report count: 1
    Packet type: Sender Report (200)
    Length: 12 (52 bytes)
    Sender SSRC: 0x055fcc01 (90164225)
    Timestamp, MSW: 2208993651 (0x83aa9173)
    Timestamp, LSW: 1584094718 (0x5e6b5dfe)
    [MSW and LSW as NTP timestamp: Jan  1, 1970 01:20:51.368825000 UTC]
    RTP timestamp: 1319200
    Sender's packet count: 250
    Sender's octet count: 33000
    Source 1
        Identifier: 0x419cbb01 (1100790529)
        SSRC contents
            Fraction lost: 0 / 256
            Cumulative number of packets lost: 0
        Extended highest sequence number received: 245
            Sequence number cycles count: 0
            Highest sequence number received: 245
        Interarrival jitter: 30
        Last SR timestamp: 0 (0x00000000)
        Delay since last SR timestamp: 0 (0 milliseconds)
Real-time Transport Control Protocol (Source description)
    [Stream setup by H245 (frame 83)]
        [Setup frame: 83]
        [Setup Method: H245]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 0001 = Source count: 1
    Packet type: Source description (202)
    Length: 3 (16 bytes)
    Chunk 1, SSRC/CSRC 0x55FCC01
        Identifier: 0x055fcc01 (90164225)
        SDES items
            Type: CNAME (user and domain) (1)
            Length: 5
            Text: AUDIO
            Type: END (0)
```

Table 4: Flow Metadata Attributes Derived from a Sample H.245 Audio Session

Flow Metadata Attributes	Values
Application Name	audio
Media Protocol	RTP
max Bit Rate (Bandwidth, in bits per second (b/s))	480
dynamicRTPPayloadType	115
Session ID	1

```
Ethernet II, Src: Viavideo_0c:96:de (00:e0:db:0c:96:de), Dst: Radvisio_01:14:93
(00:03:d6:01:14:93)
Internet Protocol Version 4, Src: 10.0.0.95 (10.0.0.95), Dst: 10.0.0.105 (10.0.0.105)
Transmission Control Protocol, Src Port: 35940 (35940), Dst Port: 39161 (39161), Seq: 664,
 H.245
    PDU Type: request (0)
        request: openLogicalChannel (3)
            openLogicalChannel
                forwardLogicalChannelNumber: 3
                forwardLogicalChannelParameters
                  dataType: videoData (2)
                   videoData: genericVideoCapability (5): ITU-T Rec. H.241 H.264 Video
Capabilities
                              genericVideoCapability
                                  capabilityIdentifier: standard (0)
                                      standard: 0.0.8.241.0.0.1 (h264 generic-capabilities)
- ITU-T Rec.
                                                H.241 H.264 Video Capabilities
                                  maxBitRate: 3360
                                  collapsing: 5 items
                        multiplexParameters: h2250LogicalChannelParameters (3)
                            h2250LogicalChannelParameters
                                sessionID: 2
                                mediaControlChannel: unicastAddress (0)
                                    unicastAddress: iPAddress (0)
                                        iPAddress
                                            network: 10.0.0.95 (10.0.0.95)
                                            tsapIdentifier: 49157
                                dynamicRTPPayloadType: 109
                                mediaPacketization: rtpPayloadType (1)
                                    rtpPayloadType
                                        payloadDescriptor: oid (2)
                                            oid: 0.0.8.241.0.0.0.0
(iPpacketization_h241AnnexA(single NAL unit mode))
                                        payloadType: 109
```

Table 5: Flow Metadata Attributes Derived from a Sample H.245 Video Session

Flow Metadata Attributes	Values
Application Name	video
Media Protocol	RTP
max Bit Rate (Bandwidth in b/s)	3360

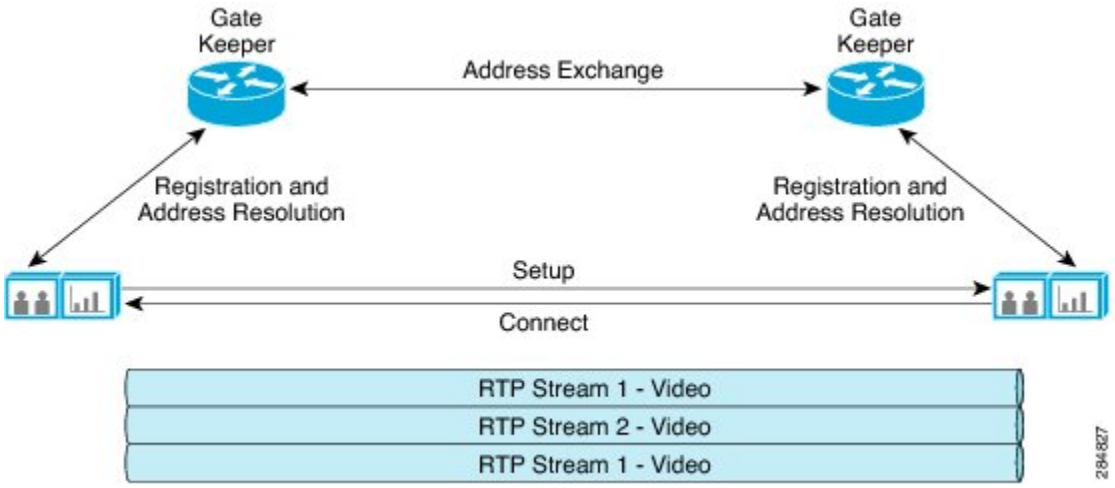
Flow Metadata Attributes	Values
dynamicRTPPayloadType	109
Video Codec	H.264
Session ID	2

H.323 Fast Connect

Fast Connect is a means of establishing an H.323 call with as few as two messages, which is achieved by tunneling H.245 messages along with H.225 messages (Setup/Connect).

Fast Connect allows endpoints to establish media channels without waiting for separate H.245 logical connections to be opened. This streamlines the number of messages that are exchanged and the amount of processing that must be done before endpoint connections can be established.

Following is an illustration of H.323-Fast Connect. The flow metadata attributes captured for Fast Connect remains similar to that of the H.323 process.

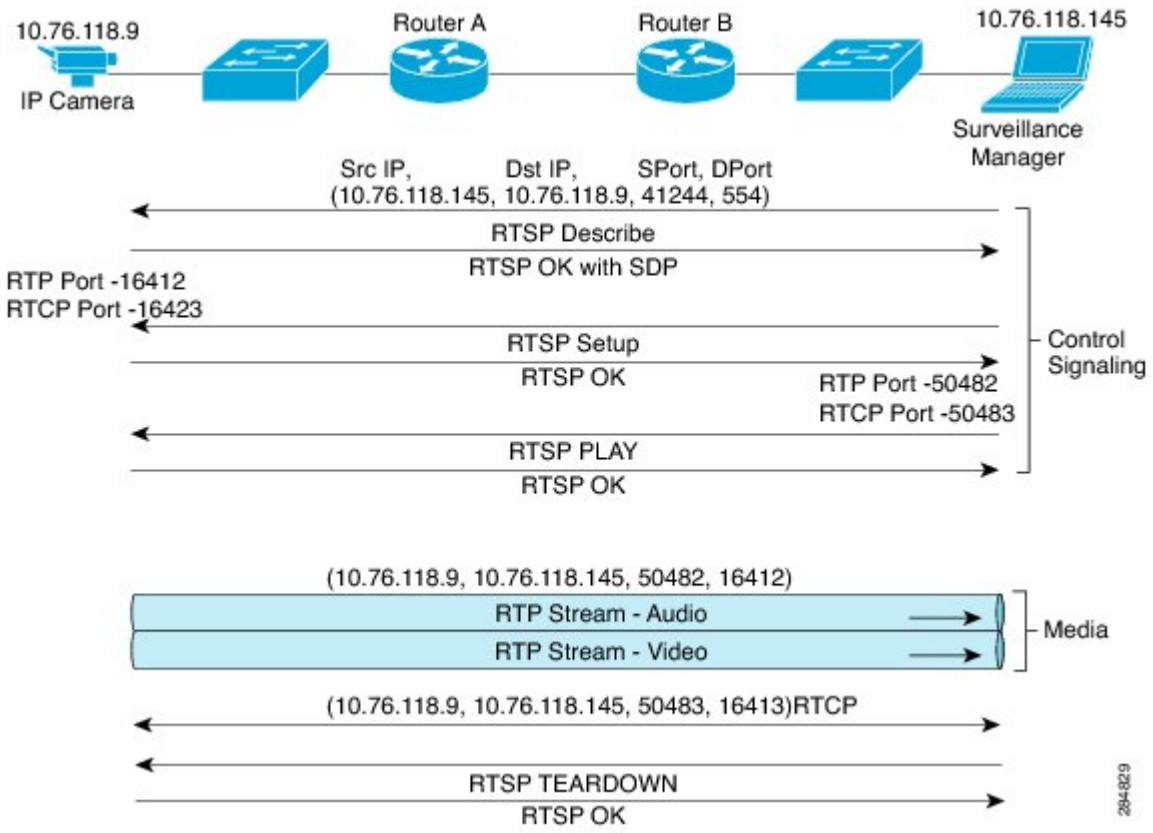


RTSP-Based Flow Identification

RTSP is an application-level protocol that provides a mechanism to control on-demand delivery of real-time data such as audio and video. It is independent of the transport protocol being used (TCP or UDP).

RTSP allows media clients to control selected, noncontiguous sections of media presentations, rendering those streams with an RTP media layer. SDP is one of the protocols used to describe streams or presentations in RTSP.

The following illustration shows the working of RTSP-based flow identification:



The client (which is the surveillance manager) sends a control request to the server (which is the IP camera), listing the source and destination IP addresses, and the source and destination port numbers.

Before establishing the session, the client must get the session description from the web server by using HTTP. The server retrieves the description of the presentation or the media object and sends the DESCRIBE message to the client. According to the information available in the description, the client sends a SETUP request to the server, specifying the transport mechanism used. The server responds to the client with an OK message along with the SDP indicating that the stream has been prepared successfully. The SDP contains the tuple values along with other flow metadata attributes that can be used to provide additional services.

The client starts the streaming (audio, video, or both) with a PLAY request and ends the streaming session with a TEARDOWN request.

The following is a sample RTSP message exchange format for flow identification:

```
Ethernet II, Src: Cisco_f0:76:76 (00:24:97:f0:76:76), Dst: AxisComm_94:12:d3
(00:40:8c:94:12:d3)
Internet Protocol Version 4, Src: 10.76.118.12 (10.76.118.12), Dst: 10.76.118.145
(10.76.118.145)
Transmission Control Protocol, Src Port: 48587 (48587), Dst Port: rtsp (554), Seq: 1, Ack:
 1, Len: 164
Real Time Streaming Protocol
    Request: DESCRIBE rtsp://10.76.118.145:554/mpeg4/1/media.amp RTSP/1.0\r\n
        Method: DESCRIBE
        URL: rtsp://10.76.118.145:554/mpeg4/1/media.amp
    CSeq: 1\r\n
    Accept: application/sdp\r\n
    Authorization: Basic YWRtaW46QyFzYzAxMjM=\r\n
    User-Agent: BroadWare\r\n
    \r\n
```

The RTSP request message starts with the method (in this case DESCRIBE), URI, and the protocol version in use:

```
Frame 120: 1011 bytes on wire (8088 bits), 1011 bytes captured (8088 bits)
Ethernet II, Src: AxisComm_94:12:d3 (00:40:8c:94:12:d3), Dst: Cisco_f0:76:76
(00:24:97:f0:76:76)
Internet Protocol Version 4, Src: 10.76.118.145 (10.76.118.145), Dst: 10.76.118.12
(10.76.118.12)
Transmission Control Protocol, Src Port: rtsp (554), Dst Port: 48587 (48587), Seq: 1, Ack:
 165, Len: 945
Real Time Streaming Protocol
    Response: RTSP/1.0 200 OK\r\n
        Status: 200
    CSeq: 1\r\n
    Content-Base: rtsp://10.76.118.145:554/mpeg4/1/media.amp/\r\n
    Content-type: application/sdp
    Content-length: 806
    \r\n
    Session Description Protocol
        Session Description Protocol Version (v): 0
        Owner/Creator, Session Id (o): - 1289587955180222 1289587955180226 IN IP4
10.76.118.145
            Owner Username: -
            Session ID: 1289587955180222
            Session Version: 1289587955180226
            Owner Network Type: IN
            Owner Address Type: IP4
            Owner Address: 10.76.118.145
        Session Name (s): Media Presentation
        E-mail Address (e): NONE
        Connection Information (c): IN IP4 0.0.0.0
            Connection Network Type: IN
            Connection Address Type: IP4
            Connection Address: 0.0.0.0
        Bandwidth Information (b): AS:8064
            Bandwidth Modifier: AS [Application Specific (RTP session bandwidth)]
            Bandwidth Value: 8064 kb/s
        Time Description, active time (t): 0 0
            Session Start Time: 0
            Session Stop Time: 0
        Session Attribute (a): control:*
            Session Attribute Fieldname: control
            Session Attribute Value: *
        Session Attribute (a): range:npt=now-
            Session Attribute Fieldname: range
            Session Attribute Value: npt=now-
        Session Attribute (a) [truncated]: mpeg4-iod:
"data:application/mpeg4-iod;base64,AoF/A8BAf7AQOEBgABQHRkXXRhcmEwcGqpY2F0aW9uL21wZWc0IDW9kIWF1C21hc2U2NCxBVGdDR3dVZkF4Y0F5U1EBWlERUklC

            RUVrK0FBZWhJQUFIb1NBQVlCQkFFWkFwOERGUUJsQlFRT1FCVUFDN2dBQV
            Session Attribute Fieldname: mpeg4-iod
            Session Attribute Value [truncated]:
"data:application/mpeg4-iod;base64,AoF/A8BAf7AQOEBgABQHRkXXRhcmEwcGqpY2F0aW9uL21wZWc0IDW9kIWF1C21hc2U2NCxBVGdDR3dVZkF4Y0F5U1EBWlERUklCQUFIb1NBQVlCQkFFWkFwOERGUUJsQlFRT1FCVUFDN2dBQVD

        Media Description, name and address (m): video 0 RTP/AVP 96
            Media Type: video
            Media Port: 0
            Media Protocol: RTP/AVP
            Media Format: DynamicRTP-Type-96
        Bandwidth Information (b): AS:8000
            Bandwidth Modifier: AS [Application Specific (RTP session bandwidth)]
            Bandwidth Value: 8000 kb/s
        Media Attribute (a): framerate:15.0
            Media Attribute Fieldname: framerate
            Media Attribute Value: 15.0
        Media Attribute (a): control:trackID=1
            Media Attribute Fieldname: control
            Media Attribute Value: trackID=1
        Media Attribute (a): rtpmap:96 MP4V-ES/90000
            Media Attribute Fieldname: rtpmap
            Media Format: 96
            MIME Type: MP4V-ES
            Sample Rate: 90000
```

The RTSP response message sent by the recipient contains the protocol version followed by the status code and the content type. The SDP contains the bandwidth information, application name, clock frequency, and other flow metadata attributes.

The following table contains the flow metadata attributes that are extracted from the sample RTSP message exchange process.

Table 6: Flow Metadata Attributes Derived from the Sample RTSP Session

Flow Metadata Attributes	Values
Application Name	video
Media Protocol	RTP
max Bit Rate (Bandwidth, in kb/s)	8064
Frame Rate	15
dynamicRTPPayloadType	96
MIME Type	MP4V-ES
Clock Frequency	90000

The SETUP request contains the RTP 5-tuple information:

```
Frame 122: 244 bytes on wire (1952 bits), 244 bytes captured (1952 bits)
Ethernet II, Src: Cisco_f0:76:76 (00:24:97:f0:76:76), Dst: AxisComm_94:12:d3
(00:40:8c:94:12:d3)
Internet Protocol Version 4, Src: 10.76.118.12 (10.76.118.12), Dst: 10.76.118.145
(10.76.118.145)
Transmission Control Protocol, Src Port: 48587 (48587), Dst Port: rtsp (554), Seq: 165,
Ack: 946, Len: 178
Real Time Streaming Protocol
    Request: SETUP rtsp://10.76.118.145:554/mpeg4/1/media.amp/trackID=1 RTSP/1.0\r\n
        Method: SETUP
        URL: rtsp://10.76.118.145:554/mpeg4/1/media.amp/trackID=1
    CSeq: 2\r\n
    Authorization: Basic YWRtaW46QyFzYzAxMjM=\r\n
    Transport: RTP/AVP/TCP;unicast
    User-Agent: BroadWare\r\n
    \r\n

Frame 123: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits)
Ethernet II, Src: AxisComm_94:12:d3 (00:40:8c:94:12:d3), Dst: Cisco_f0:76:76
(00:24:97:f0:76:76)
Internet Protocol Version 4, Src: 10.76.118.145 (10.76.118.145), Dst: 10.76.118.12
(10.76.118.12)
Transmission Control Protocol, Src Port: rtsp (554), Dst Port: 48587 (48587), Seq: 946,
Ack: 343, Len: 120
Real Time Streaming Protocol
    Response: RTSP/1.0 200 OK\r\n
        Status: 200
    CSeq: 2\r\n
    Session: 1312017293;timeout=60
    Transport: RTP/AVP/TCP;unicast;interleaved=0-1;mode="PLAY"
    \r\n
```

The PLAY request allows the RTP 5-tuple information to be extracted for further processing:

```
Frame 124: 239 bytes on wire (1912 bits), 239 bytes captured (1912 bits)
Ethernet II, Src: Cisco_f0:76:76 (00:24:97:f0:76:76), Dst: AxisComm_94:12:d3
```

```
(00:40:8c:94:12:d3)
Internet Protocol Version 4, Src: 10.76.118.12 (10.76.118.12), Dst: 10.76.118.145
(10.76.118.145)
Transmission Control Protocol, Src Port: 48587 (48587), Dst Port: rtsp (554), Seq: 343,
Ack: 1066, Len: 173
Real Time Streaming Protocol
    Request: PLAY rtsp://10.76.118.145:554/mpeg4/1/media.amp RTSP/1.0\r\n
        Method: PLAY
        URL: rtsp://10.76.118.145:554/mpeg4/1/media.amp
    CSeq: 3\r\n
    Session: 1312017293
    Authorization: Basic YWRtaW46QyFzYzAxMjM=\r\n
    Range: npt=now-\r\n
    User-Agent: BroadWare\r\n
    \r\n
```

The RTP 5-tuple values that are extracted from the sample RTSP message exchange process are listed in the table below.

Table 7: RTP Tuple Values Derived

RTP Tuple	Values
Source IP	10.76.118.12
Destination IP	10.76.118.145
Source Port	48587
Destination Port	554
Protocol	UDP
SSRC	14E59BAE
Timeout	60

The 200 OK response message indicates the data being streamed:

```
Frame 324: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits)
Ethernet II, Src: AxisComm_94:12:d3 (00:40:8c:94:12:d3), Dst: Cisco_f0:76:76
(00:24:97:f0:76:76)
Internet Protocol Version 4, Src: 10.76.118.145 (10.76.118.145), Dst: 10.76.118.9
(10.76.118.9)
Transmission Control Protocol, Src Port: rtsp (554), Dst Port: mpnjsc (1952), Seq: 1108,
Ack: 536, Len: 120
Real Time Streaming Protocol
    Response: RTSP/1.0 200 OK\r\n
        Status: 200
    CSeq: 3\r\n
    Session: 0141143570
    Range: npt=now-\r\n
    RTP-Info: url=trackID=1;seq=36491;rtptime=3364651885\r\n
    \r\n

Frame 325: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Ethernet II, Src: AxisComm_94:12:d3 (00:40:8c:94:12:d3), Dst: Cisco_f0:76:76
(00:24:97:f0:76:76)
Internet Protocol Version 4, Src: 10.76.118.145 (10.76.118.145), Dst: 10.76.118.9
(10.76.118.9)
User Datagram Protocol, Src Port: 50420 (50420), Dst Port: 16412 (16412)
    Source port: 50420 (50420)
    Destination port: 16412 (16412)
```

```
    Length: 1480
    Checksum: 0xaf6d [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Real-Time Transport Protocol
    [Stream setup by RTSP (frame 322)]
        [Setup frame: 322]
        [Setup Method: RTSP]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    Payload type: DynamicRTP-Type-96 (96)
    Sequence number: 36491
    [Extended sequence number: 36491]
    Timestamp: 3364650907
    Synchronization Source identifier: 0x3459d33f (878302015)
    Payload: 000001b0f5000001b5090000010000000120008d495880325...
```

User-Defined Port Configuration

TCP or UDP ports can be opened globally at the device level or on a per-physical port basis. The Catalyst 4500 series switches support ports only at the global level.

TCP or UDP ports can be nonstandard depending on the endpoint device. Standard ports are opened by the device by default. Users can dynamically change the port numbers using the CLI, if required.

The following table lists the standard port numbers for different protocols.

Table 8: Standard Port Numbers

Protocol	Transport Protocol	Standard Port Numbers
H.225	TCP	1720
H.323 RAS	UDP	1718
mDNS	UDP	5353
RTSP	TCP and UDP	554
SIP	TCP and UDP	5060

You can use the **profile flow port-map** command to configure a user-defined port number for the protocols.

All ports must be opened as the system gets powered up or at least before the physical port goes to the UP state. If the ports are opened after the physical port is set to UP state, the initial synch or handshake can get lost and device or flow identification can get obstructed. A link flap, or the **shutdown** command followed by the **no shutdown** command, should restart the initial handshake messages for device and flow identification.

The Catalyst 4500 switches support only one port number for a protocol. For example, if you specify 5070 as a SIP port number, the platform replaces the standard port of 5060 with 5070 in the hardware.

How to Configure Media Services Proxy

Enabling Media Services Proxy

By default, all flow identification protocols supported by MSP are enabled. If MSP is disbaled manually, perform the following task to enable MSP globally:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **profile flow**
4. **profile flow protocol** *protocol-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**profile flow** **Example:** `Device(config)# profile flow`	Enables MSP on the device.
Step 4	**profile flow protocol** *protocol-name* **Example:** `Device(config)# profile flow protocol sip`	(Optional) Enables the specified protocol. • You can use this command if any protocol is disabled manually. By default, all protocols are enabled if the profile flow command is specified.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Providing MSP Flow Services

You can provide flow services either by creating EEM scripts or by creating MSP profiles, and attaching them to each flow. MSP profiles identify the actions that must be taken on every flow. You can configure MSP profiles and customize them with flow metadata and RSVP parameters for each flow. By default, user-configured flow attributes are used by the MSP profile. When MSP is configured per interface, globally, and by using an EEM script, the order of preference is:

- Profile from the EEM script
- Profile attached to an interface
- Profile attached globally

You can provide MSP flow services using following methods:

Providing MSP Flow Services Using EEM Script

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **event manager directory user policy** *path*
4. **event manager policy** *policy-filename* **type user**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Step 2 **configure terminal**
Enters global configuration mode.

Step 3 **event manager directory user policy** *path*
Specifies a directory to be used for storing user-defined EEM policies.

Example:

```
Device(config)# event manager directory user policy flash:/policy1
```

Step 4 **event manager policy** *policy-filename* **type user**

Registers an EEM policy of a specified type and user with EEM.

Example:

```
Device(config)# event manager policy test-1-1.tcl type user
```

Providing Flow Services by Using MSP Profiles

MSP profiles identify the actions that must be taken on every flow. You can configure MSP profiles and customize them with flow metadata and RSVP parameters for each flow.

You can attach the MSP profiles to the media flow either globally or per interface.

If you attach a profile globally, RSVP and flow metadata attributes in the MSP profile are associated to all the flows identified.

If you attach a profile to an interface, RSVP and flow metadata attributes that are configured in the profile are associated with each unique flow identified on that interface.

Perform the following task to provide flow services by using MSP profiles.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **media-proxy services profile** *profile-name*
4. **rsvp**
5. **params** *rsvp-param-name*
6. **exit**
7. **metadata**
8. **params** *metadata-param-name*
9. **exit**
10. **exit**
11. **interface** *type number*
12. **media-proxy services** *profile-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**media-proxy services profile** *profile-name* **Example:** `Device(config)# media-proxy services profile profile1`	Creates an MSP profile and enters media proxy services configuration mode.
Step 4	**rsvp** **Example:** `Device(config-ms)# rsvp`	Enters media proxy services RSVP configuration mode.
Step 5	**params** *rsvp-param-name* **Example:** `Device(config-ms-rsvp)# params media-rsvp`	Associates the manually configured RSVP parameters with the MSP profile. For more information about creating RSVP parameters manually, refer to Manually Configuring RSVP CAC Parameters.
Step 6	**exit** **Example:** `Device(config-ms-rsvp)# exit`	Returns to media proxy services configuration mode.
Step 7	**metadata** **Example:** `Device(config-ms)# metadata`	Enters media proxy services metadata configuration mode.

	Command or Action	Purpose
Step 8	**params** *metadata-param-name* **Example:** `Device(config-ms-md)# paramas metadata1`	Associates the manually configured flow metadata attributes with the MSP profile. For more information about creating metadata attributes manually, refer to Manually Configuring Flow Metadata Attributes.
Step 9	**exit** **Example:** `Device(config-ms-md)# exit`	Enters media proxy services configuration mode.
Step 10	**exit** **Example:** `Device(config-ms)# exit`	Enters global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Enters interface configuration mode.
Step 12	**media-proxy services** *profile-name* **Example:** `Device(config-if)# media-proxy services profile1`	Attaches the MSP profile to the flow on the specified interface. **Note** You can attach the MSP profile globally by configuring this command in global configuration mode.
Step 13	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Manually Configuring Flow Metadata Attributes

By default, MSP identifies the endpoints and the flow by using flow identifying mechanisms and gleans the flow and device-related flow metadata attributes. You can perform the following task to manually configure flow metadata attributes. Any flow metadata attribute configured manually overrides the attribute that has been identified automatically.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **media-proxy metadata** *metadata-param-name*
4. **application name** *application-name* [**vendor***vendor-name* **version** *version-number*]
5. **bandwidth** *bw-kb/s*
6. **clock-frequency** *b/s*
7. **cname** *name*
8. **domain-name** *domain*
9. **email** *email-id*
10. **mime-type** *type*
11. **payload-type** *type*
12. **session-id** *id*
13. **ssrc** *value*
14. **username** *name*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**media-proxy metadata** *metadata-param-name* **Example:** `Device(config)# media-proxy metadata mt1`	Configures a flow metadata template and enters media proxy services metadata configuration mode.

	Command or Action	Purpose
Step 4	**application name** *application-name* [**vendor***vendor-name* **version** *version-number*] **Example:** `Device(config-ms-md)# application name app1`	Configures the name of the application, the vendor, and the version number.
Step 5	**bandwidth** *bw-kb/s* **Example:** `Device(config-ms-md)# bandwidth 1200`	Configures the bandwidth of the flow, in kb/s.
Step 6	**clock-frequency** *b/s* **Example:** `Device(config-ms-md)# clock-frequency 120`	Sets the desired clock rate, in b/s.
Step 7	**cname** *name* **Example:** `Device(config-ms-md)# cname user@example.domain.com`	Configures the canonical name. • Consists of user and domain name in one of the following formats—user@example.domain.com, user@10.10.10.1.
Step 8	**domain-name** *domain* **Example:** `Device(config-ms-md)# domain-name example.com`	Configures the domain name of the application.
Step 9	**email** *email-id* **Example:** `Device(config-ms-md)# email user@example.com`	Configures the e-mail ID of the user.
Step 10	**mime-type** *type* **Example:** `Device(config-ms-md)# mime-type MP4V-ES`	Specifies the Multipurpose Internet Mail Extensions (MIME) type of the flow.

	Command or Action	Purpose
Step 11	**payload-type** *type* **Example:** `Device(config-ms-md)# payload-type 96`	Configures the payload type for a given flow.
Step 12	**session-id** *id* **Example:** `Device(config-ms-md)# session-id 1`	Configures an identifier for the session established.
Step 13	**ssrc** *value* **Example:** `Device(config-ms-md)# ssrc 14E59BAE`	Configures the synchronization source (SSRC) value for a given flow. • Valid range is from 0 to 4294967295.
Step 14	**username** *name* **Example:** `Device(config-ms-md)# username user1`	Configures the username.
Step 15	**end** **Example:** `Device(config-ms-md)# end`	Returns to privileged EXEC mode.

Manually Configuring RSVP CAC Parameters

MSP triggers RSVP requests on behalf of the endpoints. Bandwidth reservation is performed automatically for media flow after the endpoint and flow details are detected by MSP.

You can perform the following task to manually configure RSVP CAC parameters when an RSVP CAC session is initiated by a router or a switch. Manually configured RSVP parameters override automatically detected RSVP CAC parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **media-proxy rsvp** *media-proxy services rsvp*
4. **bandwidth** *bw*
5. **max-burst** *burst-rate*
6. **peak-rate** *kb/s*
7. **priority** {**defending** *defend-value* | **premption** *prempt-value*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**media-proxy rsvp** *media-proxy services rsvp* **Example:** `Device(config)# media-proxy rsvp media-rsvp`	Configures an RSVP template and enters media proxy services RSVP configuration mode.
Step 4	**bandwidth** *bw* **Example:** `Device(config-ms-rsvp)# bandwidth 124`	Configures the bandwidth, in kb/s, to be assigned to the flow.
Step 5	**max-burst** *burst-rate* **Example:** `Device(config-ms-rsvp)# max-burst 34`	Configures the largest amount of data allowed in a flow, in kilobytes (KB).

	Command or Action	Purpose
Step 6	**peak-rate** *kb/s* **Example:** `Device(config-ms-rsvp)# peak-rate 56`	Configures the peak rate, in kb/s, for a given flow.
Step 7	**priority** {**defending** *defend-value* | **premption** *prempt-value*} **Example:** `Device(config-ms-rsvp)# priority defending 2`	Configures the defending or the preemption priority for the flow.
Step 8	**end** **Example:** `Device(config-ms-rsvp)# end`	Returns to privileged EXEC mode.

Configuring User-Defined Port Numbers for Protocols

By default, standard TCP or UDP ports are used for device and flow identification. You can perform the following task to override the standard port numbers and configure user-defined port numbers for the specified protocols.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **profile flow port-map** *protocol-name*[**tcp** | **udp**] *port-number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**profile flow port-map** *protocol-name*[**tcp** | **udp**] *port-number* **Example:** `Device(config)# profile flow port-map rtsp udp 1051`	Configures a user-defined port number by overriding the standard port number for the specified protocol.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Verifying the MSP Configuration

Use the following commands to verify the MSP configuration. You can use the show commands in any order:

SUMMARY STEPS

1. **enable**
2. **show profile flow**
3. **show profile flow statistics** [*protocol*]
4. **show profile device**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Step 2 **show profile flow**
Displays the number of flows that have been identified.

Example:
```
Device# show profile flow
```

```
Source-IP sPort Dest-IP dPort protocol Media Services profile
10.1.1.1  2000 10.2.2.2 2001  UDP      msp_service_A
10.1.1.4  3000 10.2.2.4 2001  UDP      msp_service_B
```

Step 3 **show profile flow statistics** [*protocol*]
Displays profile statistics for a given protocol to identify packet drops associated with the protocol, if any.

Example:

```
Device# show profile flow statistics

Total number of msp sessions: 4

Input Packets:

SIP   : 192

SAP   : 0   RTSP  : 0
H323  : 0   H245  : 0
```

Step 4 **show profile device**
Displays the media services device details.

Example:

```
Device# show profile device

MAC Address     Interface    Device class            Device Model            Device Vendor
==============  ==========   =============           ============            ==============
1cdf.0f76.f5f4   Gi1/44     Cisco-Device           Cisco-IP-Phone         CTS500-32(CTS-CODEC-PRIM)

00e0.db11.0089   Gi1/43     Video-Conference       Polycom-VCF             VIAVIDEO COMMUNICATIONS,
 INC.
88f0.7789.0ccd   Gi1/44      Cisco-Device           Cisco-IP-Phone-7975     Cisco IP Phone 7975
```

Configuration Examples for Media Services Proxy

Example: Providing MSP Flow Services Using EEM Scripts

The following example shows how to provide MSP flow services using EEM scripts:

```
enable
 configure terminal
  event manager directory user policy flash:/policy1
  event manager policy test-1-1.tcl type user
```

Following sample EEM script illustrates how the required services can be provided to a media flow. The aim of this script is to apply different profiles based on the type of flow and bandwidth. All audio flows with bandwidth less than 128 kps will have one profile attached to them, whereas all video flows with bandwidth more than 128 kbps will have another profile attached to them.

```
//Defines the header of the script. This determines when exactly the script will be called.

When you configure the event manager policy policy-filename type user command, only the
header is read.
```

```
This indicates that the script must be called for the event type 'add' and the flow detection

protocol 'sip'.

::cisco::eem::event_register_msp type add flow_detect_protocol sip
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

//Fetches flow tuple and flow metadata attributes, which are then stored in the array
'arr_einfo'.

#query the info reg the event
array set arr_einfo [event_reqinfo]

if {$_cerrno != 0} {
 set result [format "msp_event=%s; msp_src_ip=%i; msp_src_port=%d; msp_dest_ip=%i;
msp_dest_port=%d; msp_l4_proto=%d; msp_attr_bw=%d; \n%s" \
$_msp_event $_msp_src_ip $_msp_src_port $_msp_dest_ip $_msp_dest_port $_msp_l4_proto
$_msp_attr_bw $_cerr_str]
error $result
}
//Defines global variables in which the values set in the "arr_einfo" are stored. This is
optional.

# if query is successful
global msp_event
global msp_src_ip msp_src_port msp_dest_ip msp_dest_port msp_l4_proto msp_attr_bw
global msp_attr_clock_freq msp_attr_user_name msp_attr_email
global msp_attr_bw_cnsmd msp_attr_fl_detect_proto
global msp_attr_client_device_name msp_attr_client_device_model msp_attr_client_device_vendor
global msp_attr_server_device_name msp_attr_server_device_model msp_attr_server_device_vendor
 msp_attr_local_flow_id msp_attr_callid

//Assigns values to global variables.

set msp_event $arr_einfo(msp_event)
set msp_src_ip $arr_einfo(msp_src_ip)
set msp_src_port $arr_einfo(msp_src_port)
set msp_dest_ip $arr_einfo(msp_dest_ip)
set msp_dest_port $arr_einfo(msp_dest_port)
set msp_l4_proto $arr_einfo(msp_l4_proto)
set msp_attr_bw $arr_einfo(msp_attr_bw)
set msp_attr_clock_freq $arr_einfo(msp_attr_clock_freq)
set msp_attr_user_name $arr_einfo(msp_attr_user_name)
set msp_attr_email $arr_einfo(msp_attr_email)
set msp_attr_ssrc $arr_einfo(msp_attr_ssrc)
set msp_attr_bw_cnsmd $arr_einfo(msp_attr_bw_cnsmd)
set msp_attr_fl_detect_proto $arr_einfo(msp_attr_fl_detect_proto)
set msp_attr_client_device_name $arr_einfo(msp_attr_client_device_name)
set msp_attr_client_device_model $arr_einfo(msp_attr_client_device_model)
set msp_attr_client_device_vendor $arr_einfo(msp_attr_client_device_vendor)
set msp_attr_server_device_name $arr_einfo(msp_attr_server_device_name)
set msp_attr_server_device_model $arr_einfo(msp_attr_server_device_model)
set msp_attr_server_device_vendor $arr_einfo(msp_attr_server_device_vendor)
set msp_attr_local_flow_id $arr_einfo(msp_attr_local_flow_id)
set msp_attr_call_id $arr_einfo(msp_attr_call_id)

//Displays the values received by the script.
puts "****** Running SIP Script ******"
puts "Src ip $msp_src_ip Src port $msp_src_port dest_ip $msp_dest_ip dest_port $msp_dest_port
 l4 proto $msp_l4_proto bw $msp_attr_bw"
puts "Clock Freq: $msp_attr_clock_freq, User Name: $msp_attr_user_name, Email:
$msp_attr_email, Bw consumed: $msp_attr_bw_cnsmd, Flow detect proto:
$msp_attr_fl_detect_proto"
puts "Client Device: Name: $msp_attr_client_device_name Model: $msp_attr_client_device_model
 Vendor: $msp_attr_client_device_vendor"
puts "Server Device: Name: $msp_attr_server_device_name Model: $msp_attr_server_device_model
 Vendor: $msp_attr_server_device_vendor"

//Calls the Cisco IOS CLI.

if [catch {cli_open} result] {
```

```
error $result $errorInfo
} else {
array set cli $result
}

if [catch {cli_exec $cli(fd) "enable"}\
    result] {
error $result $errorInfo
} else {
set cmd_output $result
}

//Calls the MSP CLI based on the attributes.

if { $msp_attr_bw < 128 } {
#
    if [catch {cli_exec $cli(fd) "msp services attach $msp_attr_local_flow_id
$msp_attr_call_id audio-profile"}\
        result] {
            error $result $errorInfo
        } else {
            set cmd_output $result
        }
} else {
    if [catch {cli_exec $cli(fd) "msp services attach $msp_attr_local_flow_id
$msp_attr_call_id video-profile"}\
        result] {
            error $result $errorInfo
        } else {
            set cmd_output $result
        }
}
//Closes the CLI mode.
#
if [catch {cli_close $cli(fd) $cli(tty_id)} result] {
error $result $errorInfo
}
```

Example: Providing MSP Flow Services Using MSP Profiles

The following example shows how to provide flow services using MSP profiles on a per-interface basis. Note that you must have previously configured the RSVP parameters and metadata attributes manually in the media-rsvp and metadata1 arguments:

```
enable
 configure terminal
 media-proxy services profile profile1
  rsvp
   params media-rsvp
   exit
  metadata
   params metadata1
   exit
  exit
 interface gigabitethernet 0/1
  media-proxy services profile1
  end
```

The following example shows how to attach the MSP profile globally. Note that you must have previously configured the RSVP parameters and metadata attributes manually in the media-rsvp and metadata1 arguments:

```
enable
 configure terminal
 media-proxy services profile profile1
  rsvp
   params media-rsvp
   exit
  metadata
```

```
 params metadata1
 exit
exit
media-proxy services profile1
end
```

Example: Manually Configuring Flow Metadata Attributes

The following example shows how to manually configure flow metadata attributes of application app1, bandwidth 10,000 kb/s, payload-type 7, and session ID 23 that can be applied to a flow:

```
enable
 configure terminal
 media-proxy services metadata m1
  application name app1
  bandwidth 10000
  payload-type 7
  session-id 23
  end
```

Example: Manually Configuring RSVP Parameters

The following example shows how to manually configure RSVP parameters of bandwidth 1056 kb/s, max burst 3000, and a defending priority 2 that can be applied to a flow:

```
 enable
 configure terminal
 media-proxy services rsvp rs1
  bandwidth 1056
  max-burst 3000
  priority defending 2
  end
```

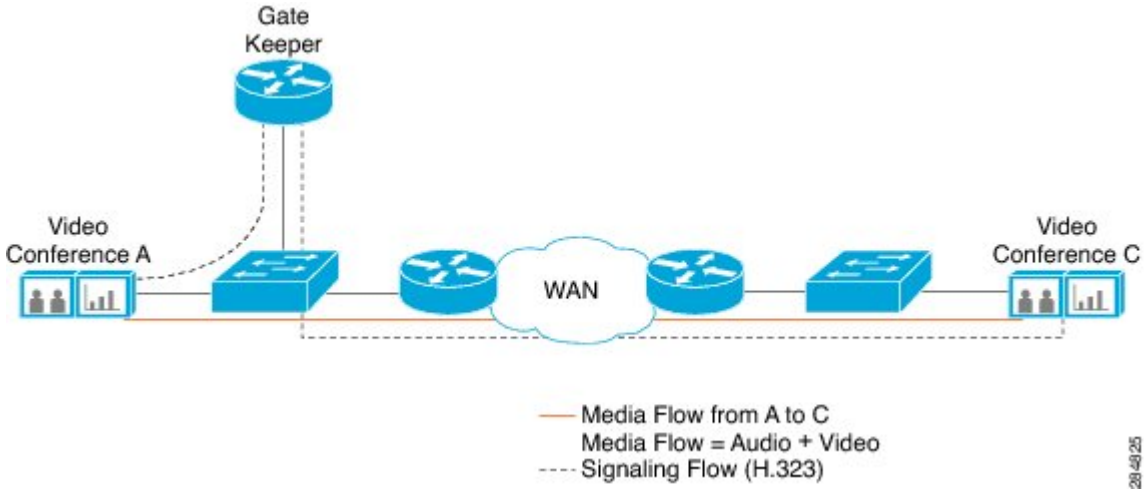
Example: Configuring User-Defined Port Numbers for Protocols

The following example shows how to configure the RTSP protocol to use port number 1051:

```
enable
 configure terminal
  profile flow port-map rtsp udp 1051
  end
```

Sample Deployment Scenario for MSP Implementation

The following section describes a video conference deployment model that uses the H.323 protocol. The illustration below provides a topology of a typical video conference system.



As depicted in the figure above, two users from two different locations are involved in a video conference through two video conference systems, Video Conference A and Video Conference C. Two networking devices, Switch A and Switch C are connected to L2 interfaces of the video conference systems.

All H.323 media register with a gatekeeper. This gatekeeper provides RAS signaling, thus achieving address resolution and admission control services.

Multiple video or audio streams can originate from these media endpoints. The video streams may have media monitoring enabled, and the Differentiated Services Code Point (DSCP) markings can be different for data and audio streams.

In this deployment model, the network operator intends to achieve the following:

- To automatically identify the H.323 flow that exceeds bandwidth of 2 Mbps and sets up QoS policy of marking to appropriate DSCP values.
- To automatically identify H.323 flow matching payload type 96, bandwidth of 64 kb/s, and an audio codec of G.711, and also to provide RSVP bandwidth reservations for the same.

Applying the following configuration on Switch A and Switch C enables MSP:

```
Device> enable
Device# configure terminal
Device(config)# profile flow
```

MSP, when enabled on Switch A and Switch C detects and identifies the type of device and the flow. Each audio or video stream is uniquely identified with the 5-tuple information (source IP, destination IP, source port, destination port, and protocol).

The following EEM script lets the system automatically identify the H.323 flow that exceeds bandwidth of 2 Mbps and sets up QoS policy of marking to appropriate DSCP values.

```
::cisco::eem::event_register_msp type add flow_detect_protocol h323

//This is the EEM script that will be executed when signaling protocol(or flow detection
protocol) is H.323 and
it is an add event
//It attaches a profile that provides RSVP services, if the bandwidth required is greater
than or equal to 2 Mbps

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

//query the info reg the event
array set arr_einfo [event_reqinfo]
```

```
if {$_cerrno != 0} {
 set result [format "msp_event=%s; msp_src_ip=%i; msp_src_port=%d; msp_dest_ip=%i;
msp_dest_port=%d; msp_l4_proto=%d;
msp_attr_bw=%d; \n%s" \
$_msp_event $_msp_src_ip $_msp_src_port $_msp_dest_ip $_msp_dest_port $_msp_l4_proto
$_msp_attr_bw $_cerr_str]
error $result
}

//if query is successful
global msp_attr_bw
global msp_attr_local_flow_id msp_attr_callid

set msp_attr_bw $arr_einfo(msp_attr_bw)
set msp_attr_local_flow_id $arr_einfo(msp_attr_local_flow_id)
set msp_attr_call_id $arr_einfo(msp_attr_call_id)
```

The following EEM script lets the system automatically the H.323 flow matching payload type 96, bandwidth of 64 kb/s, and an audio codec of G.711, and also to provide RSVP bandwidth reservations:

```
::cisco::eem::event_register_msp type add flow_detect_protocol h323

//This is the EEM script that will be executed when signaling protocol (or flow detection
protocol) is h323 and
it is an add event
//This attaches a profile that provides rsvp services, if:
- the bandwidth required is greater than or equal to 64 kbps
- payload-type is 96
- mime-type is G711

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

//query the info reg the event
array set arr_einfo [event_reqinfo]

if {$_cerrno != 0} {
 set result [format "msp_event=%s; msp_src_ip=%i; msp_src_port=%d; msp_dest_ip=%i;
msp_dest_port=%d; msp_l4_proto=%d; msp_attr_bw=%d; \n%s" \
$_msp_event $_msp_src_ip $_msp_src_port $_msp_dest_ip $_msp_dest_port $_msp_l4_proto
$_msp_attr_bw $_cerr_str]
 error $result
}

//if query is successful
global msp_attr_bw
global msp_attr_local_flow_id msp_attr_callid

set msp_attr_bw $arr_einfo(msp_attr_bw)
set msp_attr_payload_type $arr_einfo(msp_attr_payload_type)
"h323-64kbps-bw_96-pt_g711-mt.tcl" [Read only] 76 lines, 2096 characters
```

Applying the following configuration on Switch A and Switch C attaches the EEM scripts to the media flow that are automatically identified by MSP:

```
Device> enable
Device# configure terminal
Device(config)# event manager directory user policy flash:/policy1
Device(config)# event manager policy h323-2mbps-bw.tcl type user
Device(config)# event manager policy h323-64kbps-bw_96-pt_g711-mt.tcl type user
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Quality of Service Command Reference
Flow metadata overview, flow metadata properties, flow metadata entries	*Metadata Configuration Guide*

Standards and RFCs

Standard/RFC	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Media Services Proxy

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to . An account on Cisco.com is not required.

Table 9: Feature Information for Media Services Proxy

Feature Name	Releases	Feature Information
Media Services Proxy	Cisco IOS XE Release 3.3SG	MSP automatically identifies various media endpoints in the network and renders services based on the device identified. It acts as a layer that automatically connects appropriate devices with their respective network services. The following commands were introduced or modified: **media-proxy services metadata**, **media-proxy services**, **media-proxy services rsvp**, **profile flow**, **profile flow port-map**, **show profile device**, **show profile flow**.